

## **REMARKS**

This AMENDMENT UNDER 37 CFR 1.111 is filed in reply to a Notice of August 4, 2004 and the outstanding Office Action of March 19, 2004, and is believed to be fully responsive thereto for reasons set forth below in greater detail.

The Notice of Non-Compliant Amendment stated that claim 1, listed as (currently amended), did not amend the claim, which is incorrect. Refer to the third line from the end which strikes through "s" in "numbers" and adds an "s" to "slices." This was discussed with the Examiner on August 12, 2004, who requested that the AMENDMENT be resubmitted with double brackets around the "s" being deleted, which the present AMENDMENT does. Otherwise, the present AMENDMENT is substantially duplicative of the previous AMENDMENT, with the exception of this paragraph. The Notice extended the period for response for one month from August 4, 2004, so no extension fee is required.

Responsive to paragraphs 1-6 of the Office Action, the title has been amended as suggested kindly by the Examiner, the Abstract has been amended and shortened, and the specification has been amended to reference the numerals mentioned in paragraph 6 of the Office Action.

Responsive to paragraph 7 of the Office Action, a new DECLARATION has been forwarded to the inventors for signature, and will be submitted when received back from the inventors, which is expected in the near future.

Reconsideration is respectfully requested of the prior art rejections of:

Claims 1, 5, and 6 under 35 U.S.C. 103(a) as being unpatentable over Canal (Very Low Power Pipeline Using Significance Compression) in view of Emma (4,943,908);

Claims 2-4 and 8-9 under 35 U.S.C. 103(a) as being unpatentable over Canal in view of Emma as applied to claim 1 above, and further in view of Hennessy;

Claim 7 under 35 U.S.C. 103(a) as being unpatentable over Canal in view of Emma as applied to claim 1 above, and further in view of Brooks (Dynamically Exploiting Narrow Width Operands to Improve Processor Power and Performance); and

Claims 10-18 under 35 U.S.C. 103(a) as being unpatentable over Canal in view of Moline (4,941,119).

The following analysis on the applied prior art has been provided by the inventor Jude Rivers.

It appears that Emma was cited only for disclosure of somewhat conventional features. Moreover, Hennessy and Moline are cited only for subsidiary features. Accordingly, the inventor has provided the following analysis on the distinctions and advantages of the present invention over Brooks et al and Canal et al. which are considered to be more relevant to the present invention.

How does the present invention differ from Brooks et al and Canal et. al, and others and why is it better?

Brooks et al in {D. Brooks and M. Martonosi, "Dynamically exploiting narrow width operands to improve processor power and performance," Proceedings of 5th International Symposium on High-Performance Computer Architecture (HPCA-5), January 1999.} addressed this subject from the perspective of 64-bit processor implementations (the Alpha architecture, in particular), wherein the entire 64-bit width

may be required for some address computations, but such width was underutilized for most other operations. Their results show that roughly 50% of the instructions executed had both operands whose length was less than or equal to 16 bits; a large increase was found between 32 and 33-bits as a consequence of the computation of addresses for heap and stack references. Based on these results, they proposed an implementation wherein a functional unit is selectively enabled either as 16-bit or 64-bit unit, depending on the width of the operands; this determination is made dynamically, in every cycle.

Canal et al. in {R. Canal, A. Gonzalez, J. Smith, “Very low power pipelines using significance compression,” Proceedings International Conference on Microprocessors MICRO2000, Monterey, CA, December 2000.} have also recorded the presence of narrow-width operands, which they have analyzed as “significant-byte patterns.” Their study, which was based on the 32-bit MIPS architecture, shows that about 60% of all the data values used throughout the execution of the programs analyzed have only one significant byte, and 75% have at most two. Using these results, they have proposed various byte-oriented implementations that take advantage of the reduced operands width, including the ability to access only portions of the register file and the data cache.

The present invention determines the bare minimum width a-priori for executing an operation so that only the bytes required are enabled throughout the processor pipeline. In particular, the microprocessor width path is selectable before the register read stage. The operation width determination is very aggressive compared to the proposal by Brooks et al. In the Brooks et al proposal, an operation width is taken to be the wider of the two operand values. For example, in Figure 12, whereas our classification considers a) as an 8-bit wide operation, their classification would take it as a 32-bit operation. Likewise, their approach would consider the b) as a 32-bit operation while we classify it as a 16-bit operation. In both cases, our approach will do the bare minimum logic, and do a register copy or swap to fill the most significant bytes portion of the result register. In effect, we have less logic activity and conceptually consume less dynamic energy. Our effective width definition and determination is what is innovative here.

In the case of Canal et al, they do not determine an operation width a-priori. Instead, they do operations byte by byte as overflows demand. In Figure 12a for example, their approach will normally have to wait until the second cycle to realize that the operation is over. The effect here is that, even though they may save dynamic power as we do in the long run, the approach is susceptible to performance degradation. Our main objective is the reduction of dynamic power consumption with little (ideally none) performance degradation.

Based on the foregoing reasoning, our invention is substantially different in an implementation approach from, and functionally an improvement over Brooks et al. In particular, our approach to width determination and overflow detection, put together, derives a powerful solution to varying width computation that has not been shown by any prior art. The overflow detection component of our invention allows to achieve more robust savings in both power and performance, and compared to the Brooks et al. approach our invention results in power reductions by about a factor of two or better. Compared to the approach proposed by Canal et al., we believe that our approach is substantially different. Again, unlike our approach, their method does not propose a substantially new architectural implementation from the traditional microprocessor pipeline. Instead, they propose a single narrow width architecture that allows for multiple cycle execution for a wider-width computation. As much as this approach may help with some power reductions for narrow width computations, one can argue strongly that there is a potential performance bottleneck in situations with wider-width operations.

Based on the foregoing analysis, independent claim 1 distinguishes over the prior art by the recited limitations in the second and third paragraphs, independent claim 10 distinguishes over the prior art by the recited limitations in the third through sixth paragraphs, and independent claim 17, distinguishes over the prior art by the limitations recited in all of its paragraphs.

This application is now believed to be in condition for allowance, and a Notice of Allowance is respectfully requested. If the Examiner believes a telephone conference might expedite prosecution of this case, it is respectfully requested that he call applicant's attorney at (516) 742-4343.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "William C. Roch".

William C. Roch  
Registration No. 24,972

SCULLY, SCOTT, MURPHY & PRESSER  
400 Garden City Plaza  
Garden City, New York 11530  
(516) 742-4343

WCR/jf